N-Queens Problem Solving Using Linear Programming in GNU Linear Programming Kit (GLPK).

Prepared by: Mohammed Abdullah Al-Rudaini

Tools Used:

1. **GNU Linear Programming Kit: GLPK Version 4.60**

Part 1: Normal N-Queens Problem:

Given:

1. N number of Queens, 16 is used: n=16.
2. N*N Chess Board, used 16*16 Binary Matrix, Position x,y is set to 1 if a queen is placed on it:X[n,n]

Rules:

1. Total of n queens:
$$\sum_{i=1}^{n}\sum_{j=1}^{n} X[i.j] = n.$$
2. At most one queen per column:
$$\forall i = 1\ to\ n\ \sum_{j=1}^{n} X[i.j] <= 1.$$
3. At most one queen per row:
$$\forall j = 1\ to\ n\ \sum_{i=1}^{n} X[i.j] <= 1.$$
4. At most one queen per positive diagonal:
$$\forall\ k = 2-n\ to\ n-2 \sum_{i.j\in 1\ to\ n.\ i-j=k} X[i.j] <= 1$$
5. At most one queen per negative diagonal:
$$\forall\ k = 3\ to\ 2n-1 \sum_{i.j\in 1\ to\ n.\ i+j=k} X[i.j] <= 1$$

Model Code:

```
param n, integer, > 0, default 16;
/* size of the chess board */
var x{1..n, 1..n}, binary;
/* x[i,j] = 1 means that a queen is placed in square [i,j] */
s.t. a: sum{i in 1..n} sum{j in 1..n} x[i,j] == n;
/*total of n Queens*/
s.t. b{i in 1..n}: sum{j in 1..n} x[i,j] <= 1;
/* at most one queen can be placed in each row */
s.t. c{j in 1..n}: sum{i in 1..n} x[i,j] <= 1;
/* at most one queen can be placed in each column */
s.t. d{k in 2-n..n-2}: sum{i in 1..n, j in 1..n: i-j == k} x[i,j] <= 1;
/* at most one queen can be placed in each "\"-diagonal */
s.t. e{k in 3..n+n-1}: sum{i in 1..n, j in 1..n: i+j == k} x[i,j] <= 1;
/* at most one queen can be placed in each "/"-diagonal */
/* solve the problem */
solve;
/* and print its optimal solution */
for {i in 1..n}
{  for {j in 1..n} printf " %s", if x[i,j] then "Q" else ".";
   printf("\n");
}
end;
```

Solving the Model:

```
D:\master\AI\Assignments\5\Assignment5\models>glpsol --model NQ-GLPK1.mod
GLPSOL: GLPK LP/MIP Solver, v4.60
Parameter(s) specified in the command line:
 --model NQ-GLPK1.mod
Reading model section from NQ-GLPK1.mod...
NQ-GLPK1.mod:31: warning: final NL missing before end of file
31 lines were read
Generating a...
Generating b...
Generating c...
Generating d...
Generating e...
Model has been successfully generated
GLPK Integer Optimizer, v4.60
91 rows, 256 columns, 1276 non-zeros
256 integer variables, all of which are binary
Preprocessing...
91 rows, 256 columns, 1276 non-zeros
256 integer variables, all of which are binary
Scaling...
 A: min|aij| =  1.000e+00  max|aij| =  1.000e+00  ratio =  1.000e+00
Problem data seem to be well scaled
Constructing initial basis...
Size of triangular part is 91
Solving LP relaxation...
GLPK Simplex Optimizer, v4.60
91 rows, 256 columns, 1276 non-zeros
      0: obj =   0.000000000e+00 inf =   6.000e+01 (4)
     67: obj =   0.000000000e+00 inf =   8.601e-16 (0)
OPTIMAL LP SOLUTION FOUND
Integer optimization begins...
+    67: mip =     not found yet >=              -inf        (1; 0)
+   276: >>>>>   0.000000000e+00 >=   0.000000000e+00   0.0% (20; 0)
+   276: mip =   0.000000000e+00 >=     tree is empty   0.0% (0; 39)
INTEGER OPTIMAL SOLUTION FOUND
Time used:   0.1 secs
Memory used: 0.5 Mb (493157 bytes)
. . . . . Q . .
. . . . Q . . . . . . . . .
. . . . . Q . . . . . . . .
Q . . . . . . . . . . . . . .
. . Q . . . . . . . . . . . .
. . . . . Q . . . . . . . .
. . . . . Q . . . . . . . .
. . . . . Q
. . . Q . . . . . . . . . . .
. Q . . . . . . . . . . . . .
. . . . . Q . . . .
. . . . . Q . . . . . . . . .
. . . . . Q .
. . . . . Q . . .
. . . . . Q . . . . . .
. . . . . Q . . . . . . .
Model has been successfully processed
```

Part 2: N/2 Queens and N/2 Rocks:

Given:

1. n/2 number of Queens, 8 is used: n=16.
2. n/2 number of Rocks, 8 is used: n=16.
3. n*n Chess Board, used **two** 16*16 Binary Matrix, Position x,y is set to 1 if a queen is placed on X[n,n], Position x,y is set to 1 if a rock is placed on Y[n,n]

Rules:

1. Total of $\lceil n/2 \rceil$ queens:
$$\sum_{i=1}^{n} \sum_{j=1}^{n} X[i.j] = \lceil n/2 \rceil.$$
2. Total of $\lfloor n/2 \rfloor$ rocks:
$$\sum_{i=1}^{n} \sum_{j=1}^{n} Y[i.j] = \lfloor n/2 \rfloor.$$
3. The $\lceil n/2 \rceil$queens are placed in the odd columns:
$$\sum_{j \in 1 \; to \; n. \;\; j \; mod \; 2=1} X[i.j] = \lceil n/2 \rceil.$$
4. The $\lfloor n/2 \rfloor$ rocks are placed in the even columns:
$$\sum_{j \in 1 \; to \; n. \;\; j \; mod \; 2=0} Y[i.j] = \lfloor n/2 \rfloor.$$
5. At most one queen or rock per column:
$$\forall i = 1 \; to \; n \;\; \sum_{j=1}^{n} X[i.j] + \sum_{j=1}^{n} Y[i.j] \leq 1.$$
6. At most one queen or rock per row:
$$\forall j = 1 \; to \; n \;\; \sum_{i=1}^{n} X[i.j] + \sum_{i=1}^{n} Y[i.j] \leq 1.$$
7. At most one queen per positive diagonal:
$$\forall \; k = 2 - n \; to \; n - 2 \qquad \sum_{i.j \in 1 \; to \; n. \;\; i-j=k} X[i.j] \leq 1$$
8. 0 to $\lfloor n/2 \rfloor$ rocks per positive diagonal:
$$\forall \; k = 2 - n \; to \; n - 2 \qquad \sum_{i.j \in 1 \; to \; n. \;\; i-j=k} Y[i.j] \leq \lfloor n/2 \rfloor$$
9. At most one queen with no rock or no queen with 0 to $\lfloor n/2 \rfloor$ rocks per positive diagonal:
$$\forall \; k = 2 - n \; to \; n - 2 \left(\lfloor n/2 \rfloor * \left(1 - \sum_{i.j \in 1 \; to \; n. \;\; i+j=k} X[i.j] \right) \right) \geq \left(\sum_{i.j \in 1 \; to \; n. \;\; i+j=k} Y[i.j] \right)$$
10. At most one queen per negative diagonal:
$$\forall \; k = 3 \; to \; 2n - 1 \qquad \sum_{i.j \in 1 \; to \; n. \;\; i+j=k} X[i.j] \leq 1$$
11. 0 to $\lfloor n/2 \rfloor$ rocks per negative diagonal:
$$\forall \; k = 3 \; to \; 2n - 1 \qquad \sum_{i.j \in 1 \; to \; n. \;\; i+j=k} Y[i.j] \leq \lfloor n/2 \rfloor$$
12. At most one queen with no rock or no queen with 0 to $\lfloor n/2 \rfloor$ rocks per negative diagonal:
$$\forall \; k = 3 \; to \; 2n - 1 \left(n * \left(1 - \sum_{i.j \in 1 \; to \; n. \;\; i+j=k} X[i.j] \right) \right) \geq \left(\sum_{i.j \in 1 \; to \; n. \;\; i+j=k} Y[i.j] \right)$$

Model Code:

```
param n, integer, > 0, default 16;
/* size of the chess board */
var x{1..n, 1..n}, binary;
/* x[i,j] = 1 means that a queen is placed in square [i,j] */
var y{1..n, 1..n}, binary;
/* y[i,j] = 1 means that a rock is placed in square [i,j] */
s.t. a: sum{i in 1..n} sum{j in 1..n} x[i,j] == ceil(n/2);
/*total of ceil(n/2) Queens*/
s.t. b: sum{i in 1..n} sum{j in 1..n} y[i,j] == floor(n/2);
/*total of n/2 Rocks*/
s.t. c: sum{i in 1..n} sum{j in 1..n: j mod 2 = 1} x[i,j] == ceil(n/2);
/*The ceil(n/2) Queens Placed in odd Columns*/
s.t. d: sum{i in 1..n} sum{j in 1..n: j mod 2 = 0} y[i,j] == floor(n/2);
/*The floor(n/2) Rocks Placed in even Column*/
s.t. e{i in 1..n}: (sum{j in 1..n} x[i,j] + sum{j in 1..n} y[i,j])<= 1;
/* at most one queen or rock can be placed in each row */
s.t. f{j in 1..n}: (sum{i in 1..n} x[i,j] + sum{i in 1..n} y[i,j]) <= 1;
/* at most one queen or rock can be placed in each column */
s.t. g{k in 2-n..n-2}: sum{i in 1..n, j in 1..n: i-j == k} x[i,j] <= 1;
/* at most one queen can be placed in each "\"-diagonal */
s.t. h{k in 2-n..n-2}: sum{i in 1..n, j in 1..n: i-j == k} y[i,j] <=
floor(n/2);
/* 0 to floor(n/2) rocks can be placed in each "\"-diagonal */
s.t. l{k in 2-n..n-2}: floor(n/2)*(1-sum{i in 1..n, j in 1..n: i-j == k}
x[i,j]) >= sum{i in 1..n, j in 1..n: i-j == k} y[i,j];
/* one queen with no rock or no queen with 0 to floor(n/2) rocks can be
placed in each "\"-diagonal */
s.t. m{k in 3..n+n-1}: sum{i in 1..n, j in 1..n: i+j == k} x[i,j] <= 1;
/* at most one queen can be placed in each "/"-diagonal */
s.t. o{k in 3..n+n-1}: sum{i in 1..n, j in 1..n: i+j == k} y[i,j] <=
floor(n/2);
/* 0 to floor(n/2) rocks can be placed in each "/"-diagonal */
s.t. p{k in 3..n+n-1}: floor(n/2)*(1-sum{i in 1..n, j in 1..n: i+j == k}
x[i,j]) >= sum{i in 1..n, j in 1..n: i+j == k} y[i,j];
/*one queen with no rock or no queen with 0 to floor(n/2) rocks can be
placed in each "/"-diagonal */
/* solve the problem */
solve;
/* and print its optimal solution */
for {i in 1..n}
{  for {j in 1..n} printf " %s", if x[i,j] then "Q" else if y[i,j] then
"R" else ".";
   printf("\n");
}
end;
```

Solving the Model:

```
D:\master\AI\Assignments\5\Assignment5\models>glpsol -m NQ-GLPK2.mod
GLPSOL: GLPK LP/MIP Solver, v4.60
Parameter(s) specified in the command line:
 -m NQ-GLPK2.mod
Reading model section from NQ-GLPK2.mod...
NQ-GLPK2.mod:54: warning: final NL missing before end of file
54 lines were read
Generating a...
Generating b...
Generating c...
Generating d...
Generating e...
Generating f...
Generating g...
Generating h...
Generating l...
Generating m...
Generating o...
Generating p...
Model has been successfully generated
GLPK Integer Optimizer, v4.60
210 rows, 512 columns, 3824 non-zeros
512 integer variables, all of which are binary
Preprocessing...
4 hidden covering inequaliti(es) were detected
182 rows, 512 columns, 3684 non-zeros
512 integer variables, all of which are binary
Scaling...
 A: min|aij| =  1.000e+00  max|aij| =  8.000e+00  ratio =  8.000e+00
Problem data seem to be well scaled
Constructing initial basis...
Size of triangular part is 182
Solving LP relaxation...
GLPK Simplex Optimizer, v4.60
182 rows, 512 columns, 3684 non-zeros
      0: obj =   0.000000000e+00 inf =   1.690e+02 (9)
     68: obj =   0.000000000e+00 inf =   1.085e-14 (0)
OPTIMAL LP SOLUTION FOUND
Integer optimization begins...
+    68: mip =     not found yet >=              -inf        (1; 0)
+   392: >>>>>   0.000000000e+00 >=   0.000000000e+00   0.0% (16; 1)
+   392: mip =   0.000000000e+00 >=     tree is empty   0.0% (0; 33)
INTEGER OPTIMAL SOLUTION FOUND
Time used:   0.1 secs
Memory used: 1.0 Mb (1060889 bytes)
 . . . . . R . .
 . . . . . R . . . .
 . . . . . Q . . . . . .
 Q . . . . . .
 . . . . Q . . . . . .
 . . . . . R
 . . . . . R . . . . .
 . . . . . R . . . . . .
 . . . . . Q . . . . .
 . . . . . R . . . . . .
 . R . . . . . .
 . . . . . Q . . .
 . . Q . . . . . .
 . . . . . Q . . . . . .
 . . . R . . . . . . .
 . . . . . Q .
Model has been successfully processed
```